

## Chapter 3

# The Abstraction

This chapter discussed the first issue that is highlighted in this research when we want to perform learning which is what the appropriate representation of transferred knowledge is. The type of knowledge that is transferred between source and target tasks in this research is policy. An abstract policy that is extracted from a learned policy of a source task by abstraction process is introduced as transferred knowledge. A modified learning vector quantization (LVQ) algorithm that can autonomously increase or decrease as required the number of its network neurons is proposed to perform this abstraction. Simulation results show that the transferred knowledge represented by abstract policy has fewer data and simple enough to be interpreted and that the transfer successfully improves the learning in the target task.

### 3.1 Introduction

In order to guarantee the efficiency of transfer learning, one of the factors that need to be considered is what the appropriate representation of transferred knowledge is [15]. The representation can range from very low-level information about a specific task to general heuristics that attempt to guide learning. Depending on how similar the tasks, different representation of may cause positive or negative transfer. For example, low-level information may transfer across closely related tasks, while high-level concepts may transfer across pairs of less similar tasks. Based on the type of transfer knowledge and how similar the source and target tasks, an appropriate representation is need to be found.

In this research, a high-level representation of transferred knowledge is proposed. It is expected to work as good as the low-level representation for closely related tasks

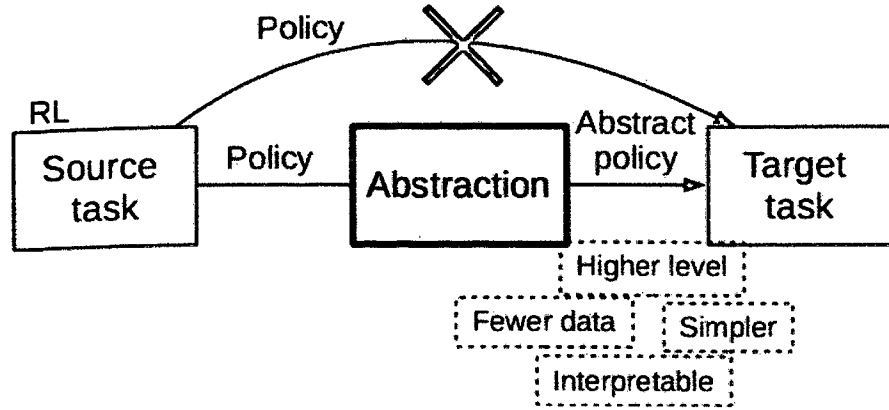


FIGURE 3.1: Policy abstraction for transfer learning.

and better for less similar tasks. As shown in Figure 3.1, instead of transferring the ordinary learned policy, rules that extracted through abstraction from the learned policy is transferred. The generated rules are expected to have fewer data compared to ordinary policy and simple enough to be interpreted. In this research, a modified learning vector quantization (LVQ) algorithm that can autonomously add or delete its network neurons is proposed to perform this abstraction and extracted rules is called abstract policy.

If the source and target tasks are very different, transfer learning might not be work so well. While allowing transfer to happen between less similar source and target tasks gives more flexibility to a designer. In this research, the tasks are similar in the terms of the tasks' objective, action set, environment objects and the number of state's variables. While tasks are different in the terms of possible states caused by the different settings of environment objects.

The rest of this chapter is organized as follows. In the Section 3.2, the idea behind the abstraction and the proposed LVQ algorithm is described. Then, it follows by the detailed explanation of the LVQ algorithm. The simulations and their results are explained in Section 3.4. Finally, Section 3.5 states the summary.

In simulation, a 3-D maze problem with a camera-mounted agent is employed. The agent is trained to move from the start state towards the goal state by avoiding some obstacles. The results show that the abstraction is successful and the abstract policy represented by weight vectors is simple and easy to interpret.

The rest of this chapter is organized as follows. In the Section 3.2, the issues and solution are described. Then, it follows by the detailed explanation of the proposed algorithm that were used in this paper. The simulations and their results are explained in Section 3.4. Finally, Section 3.5 states the summary.

### 3.2 Learning Vector Quantization (LVQ) as an Abstraction Method

In this chapter, the source task is trained using Q-learning which is one of the conventional RL methods. After the training completed, a learned policy is obtained. Here, instead of being transfer directly to the target task, the learned policy is abstracted using a modified LVQ and abstract policy is generated.

Q-learning uses lookup table to represent its policy which is actually the state-action values. As shown in Figure 3.2a, the table size is  $N \times M$ , where  $N$  is the the number of different possible states, and  $M$  is the number of different possible actions. During action selection for a certain state, agent refers to the table and lookup corresponding action values for that state, and choose the maximum. The agent will evaluate the state-action values repeatedly or in other words update its policy during learning. As illustrated in Figure 3.2b, after learning complete, the agent will have a learned policy that has the ideal actions for all possible states. It is also means that after the learning finished, all possible states are classified to the number of actions set, e.g. three classes.

Abstraction is an operation that reduce the complexity of a problem by ignoring irrelevant properties while preserving all the important ones necessary to still be able solve a given problem. In this research, the abstraction is performed by grouping different states that correspond same actions. As shown in Figure 3.3, all states that are classified to  $M$  classes by Q-learning are re-classified by abstraction to several more subclasses. Each subclass has its own ideal action.

The classification is performed based on the states' values and ideal actions. Here, during abstraction, the essential properties that are preserved are the ideal actions, while the less relevant information are the states' values. The states that are close to each other and

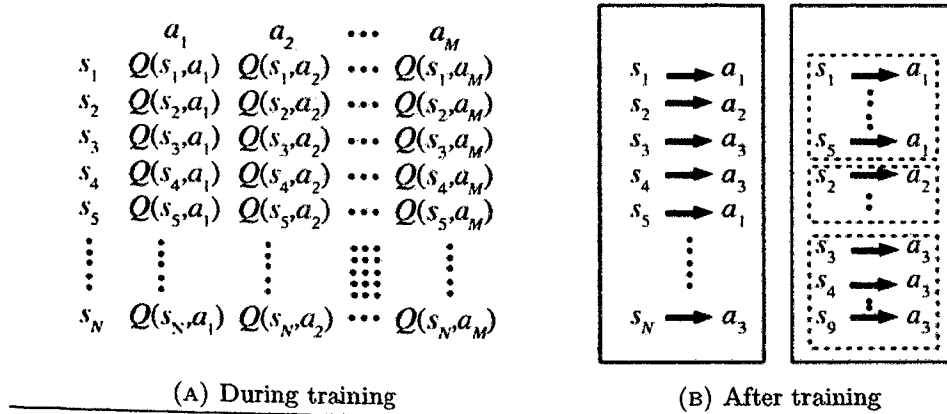


FIGURE 3.2: Policy representation using lookup table